EMERGENCE OF RANDOMNESS FROM CHAOS

René Lozi

Published in
International Journal of Bifurcation and Chaos,
Vol. 22, No. 2 (2012) pp. 1250021-1 to 1250021-15
DOI: 10.1142/S0218127412500216

EMERGENCE OF RANDOMNESS FROM CHAOS

R. LOZI
Laboratoire J.A. Dieudonné, UMR CNRS 7351,
Université de Nice Sophia-Antipolis, Parc Valrose,
06108 NICE Cedex 02, France
I.U.F.M. Université de Nice Sophia-Antipolis,
89, Avenue George V, 06046 Nice Cedex 1, France
lozi@unice.fr

Received December 15, 2011

In systems theory and science, emergence is the way complex systems and patterns arise out of a multiplicity of relatively simple interactions. Emergence is central to the theories of integrative levels and of complex systems [Aziz-Alaoui & Bertelle, 2009]. In this paper, we use the emergent property of the ultra weak multidimensional coupling of p 1-dimensional dynamical chaotic systems which leads from chaos to randomness.

Generation of random or pseudorandom numbers, nowadays, is a key feature of industrial mathematics. Pseudorandom or chaotic numbers are used in many areas of contemporary technology such as modern communication systems and engineering applications. More and more European or US patents using discrete mappings for this purpose are obtained by researchers of discrete dynamical systems [Petersen & Sorensen, 2007; Ruggiero *et al.*, 2006]. Efficient Chaotic Pseudo Random Number Generators (CPRNG) have been recently introduced. They use the ultra weak multidimensional coupling of p 1-dimensional dynamical systems which preserve the chaotic properties of the continuous models in numerical experiments. Together with chaotic sampling and mixing processes, ultra weak coupling leads to families of (CPRNG) which are noteworthy [Hénaff *et al.*, 2009a, 2009b, 2009c, 2010].

In this paper we improve again these families using a double threshold chaotic sampling instead of a single one.

We analyze numerically the properties of these new families and underline their very high qualities and usefulness as CPRNG when very long series are computed. Moreover, a determining property of such improved CPRNG is the high number of parameters used and the high sensitivity to the parameters value which allows choosing it as cipher-keys. It is why we call these families multiparameter chaotic pseudo-random number generators (M-p CPRNG).

Keywords: Emergence; randomness; chaos; discrete time systems; floating point arithmetic; random number generation.

1. Introduction

Efficient Chaotic Pseudo Random Number Generators (CPRNG) have been recently introduced. The idea of applying discrete chaotic dynamical systems, intrinsically, exploits the property of extreme sensitivity of trajectories to small changes of initial conditions. They use the ultra weak multidimensional coupling of p 1-dimensional dynamical systems which preserve the chaotic properties of the continuous models in numerical experiments. The process of chaotic sampling and mixing of chaotic sequences, which is pivotal for these families, works perfectly in numerical simulation when floating point (or double precision) numbers are handled by a computer.

It is noteworthy that these families of very weakly coupled maps are more powerful than the usual formulas used to generate chaotic sequences mainly because only additions and multiplications are used in the computation process; no division being required. Moreover, the computations are done using floating point or double precision numbers, allowing the use of the powerful Floating Point Unit (FPU) of the modern microprocessors (built by both Intel and Advanced Micro Devices (AMD)). In addition, a large part of the computations can be parallelized taking advantage of the multicore microprocessors which appear on the market of laptop computers.

In this paper we improve the properties of these families using a double threshold chaotic sampling instead of a single one. The genuine map f used as one-dimensional dynamical systems to generate them is henceforth perfectly hidden.

A determining property of such improved CPRNG is the high number of parameters used ($p \times (p-1)$ for p coupled equations) which allows to choose it as cipher-keys due to the high sensitivity to the parameter values. This is why we call these families multiparameter chaotic pseudo-random number generators (M-p CPRNG).

Several applications can be found for these families, as for example, producing Gaussian noise, computing hash function or in chaotic cryptography.

In Sec. 2 we define the double threshold chaotic sampling, in Sec. 3 we describe the emergence of randomness in a particular window of parameter value. We point out the parameter sensitivity in Sec. 4, with some applications of the M-p CPRNG. Finally in Appendix A we recall some basic properties of the previous CPRNG which allow the use of the double threshold chaotic sampling.

2. Multiparameter Chaotic Pseudo-Random Number Generator (M-p CPRNG)

When a dynamical system is realized on a computer using floating point or double precision numbers, the computation is of a discretization, where finite machine arithmetic replaces continuum state space. For chaotic dynamical systems, the discretization often has collapsing effects to a fixed point or to short cycles [Lanford III, 1998; Gora *et al.*, 2006]. In order to preserve the chaotic properties of the continuous models in numerical experiments we consider an ultra weak multidimensional coupling of p 1-dimensional dynamical systems.

2.1. *System of p-coupled symmetric tent map*

In order to simplify the presentation of the M-p CPRNG we introduce, we use as an example the symmetric tent map defined by

$$f_a(x) = 1 - a|x| \tag{1}$$

with the parameter value $a = 2$, later denoted simply as f, even though other chaotic maps of the interval (as the logistic map, the baker transform) can be used for the same purpose (as a matter of course, the invariant measure of the chaotic map chosen is preserved). The dynamical system associated to this one-dimensional map is defined on the interval $[-1;1] \subset \mathbb{R}$ [Sprott, 2003] by the equation:

$$x_{n+1} = 1 - a|x_n|. \tag{2}$$

The considered system of the p-coupled dynamical systems is described by:

$$X_{n+1} = F(X_n) = A.(\underline{f}(X_n)) \tag{3}$$

with

$$X_n = \begin{pmatrix} x_n^1 \\ \vdots \\ x_n^p \end{pmatrix} \quad \underline{f}(X_n) = \begin{pmatrix} f(x_n^1) \\ \vdots \\ f(x_n^p) \end{pmatrix} \tag{4}$$

and

$$A = \begin{pmatrix} \epsilon_{1,1} = 1 - \sum_{j=2}^{j=p} \epsilon_{1,j} & \epsilon_{1,2} & \cdots & \epsilon_{1,p-1} & \epsilon_{1,p} \\ \epsilon_{2,1} & \epsilon_{2,2} = 1 - \sum_{j=1,j\neq 2}^{j=p} \epsilon_{2,j} & \cdots & \epsilon_{2,p-1} & \epsilon_{2,p} \\ \vdots & \ddots & & \vdots & \vdots \\ \vdots & & \ddots & \vdots & \vdots \\ \epsilon_{p,1} & \cdots & \cdots & \epsilon_{p,p-1} & \epsilon_{p,p} = 1 - \sum_{j=1}^{j=p-1} \epsilon_{p,j} \end{pmatrix} \tag{5}$$

F is a map of $J^p = [-1, 1]^p \subset \mathbb{R}^p$ into itself.

Considering

$$\epsilon_{i,i} = 1 - \sum_{j=1,j\neq i}^{j=p} \epsilon_{i,j},$$

the matrix A is always a stochastic matrix iff the coupling constants verify $\epsilon_{i,j} > 0$ for every i and j.

If $\epsilon_{i,j} = 0$, for $i \neq j$, the maps are totally decoupled, whereas they are fully crisscross coupled when for example, $\epsilon_{i,j} = \frac{1}{p-1}$, for $i \neq j$. Generally, researchers do not consider very small values of $\epsilon_{i,j}$ because it seems that the maps are quasi-decoupled with those values and no special effect of the coupling is expected. In fact, it is not the case and ultra small coupling constants (as small as 10^{-7} for floating point numbers or 10^{-16} for double precision numbers) allow the construction of very long periodic orbits, leading to sterling chaotic generators. In this way, the randomness emerges from chaos.

Moreover, each component of these numbers belonging to $\mathbb{R}^p$ is equally distributed over the finite interval $J \subset \mathbb{R}$, when one chooses a function f with uniform invariant measure. Numerical computations (up to 10^{13} numbers) show that this distribution is obtained with a very good approximation. They also have the property that the length of the periods of the numerically observed orbits is very large [Lozi, 2006].

2.2. *Chaotic sampling and mixing*

However, chaotic numbers are not pseudo-random numbers because the plot of the couples of any component (x_n^l, x_{n+1}^l) of iterated points (X_n, X_{n+1}) in the corresponding phase plane reveals the map f used as one-dimensional dynamical systems to generate them via Eq. (3).

Nevertheless, we have recently introduced a family of enhanced Chaotic Pseudo Random Number Generators (CPRNG) in order to faster compute long series of pseudorandom numbers with desktop computer [Lozi, 2008a, 2008b]. This family is based on the previous ultra weak coupling which is improved in order to conceal the chaotic genuine function.

In order to hide f in the phase space (x_n^l, x_{n+1}^l) two mechanisms are used. The pivotal idea of the first one mechanism is to sample chaotically the sequence $(x_0^l, x_1^l, x_2^l, \ldots, x_n^l, x_{n+1}^l, \ldots)$ generated by the lth component x^l, selecting x_n^l every time the value x_n^m of the mth component x^m, is strictly greater (or smaller) than a threshold $T \in J$, with $l \neq m$, for $1 \leq l, m \leq p$.

That is to say, to extract the subsequence $(x_{n_{(0)}}^l, x_{n_{(1)}}^l, x_{n_{(2)}}^l, \ldots, x_{n_{(q)}}^l, x_{n_{(q+1)}}^l, \ldots)$ denoted here $(\overline{x_0}, \overline{x_1}, \overline{x_2}, \ldots, \overline{x_q}, \overline{x_{q+1}}, \ldots)$ of the original one, in the following way.

Given $1 \leq l, m \leq p$, $l \neq m$

$$\begin{cases} n_{(-1)} = -1 \\ \overline{x_q} = x_{n_{(q)}}^l, \quad \text{with } n_{(q)} = \min\limits_{r\in\mathbb{N}} \{r > n_{(q-1)} \,|\, x_r^m > T\} \end{cases} \tag{6}$$

The sequence $(\overline{x_0}, \overline{x_1}, \overline{x_2}, \ldots, \overline{x_q}, \overline{x_{q+1}}, \ldots)$ is then the sequence of chaotic pseudo-random numbers.

The mathematical formula (6) can be best understood in algorithmic way. The pseudo-code, for computing iterates of (6) corresponding to N iterates of (3) is:

$X_0 = (x_0^1, x_0^2, \ldots, x_0^{p-1}, x_0^p) = \text{seed}$
$n = 0;\ q = 0;$
do { while $n < N$
do { while $(x_n^m \leq T)$
compute $(x_n^1, x_n^2, \ldots, x_n^{p-1}, x_n^p); n{+}{+}\}$
compute $(x_n^1, x_n^2, \ldots, x_n^{p-1}, x_n^p);$
then $n(q) = n; \overline{x_q} = x_{n(q)}^1; n{+}{+}; q{+}{+}\}$

This chaotic sampling is possible due to the independence of each component of the iterated points X_n versus the others (see Appendix A.1).

Remark 2.1. Albeit the number $N\,\mathrm{Sampl}_{\mathrm{iter}}$ of pseudo-random numbers $\overline{x_q}$ corresponding to the computation of N iterates is not known *a priori*, considering that the selecting process is again linked to the uniform distribution of the iterates of the tent map on J, this number is equivalent to $\frac{2N}{1-T}$.

A second mechanism can improve the unpredictability of the pseudo-random sequence generated as above, using synergistically all the components of the vector X_n, instead of two. Given $p-1$ thresholds

$$T_1 < T_2 < \cdots < T_{p-1} \in J \tag{7}$$

and the corresponding partition of

$$J = \bigcup_{k=0}^{p-1} J_k \tag{8}$$

with $J_0 = [-1, T_1]$, $J_1 = \,]T_1, T_2[$, $J_k = [T_k, T_{k-1}[$ for $1 < k < p-1$ and $J_{p-1} = [T_{p-1}, 1[$, this simple mechanism is based on the chaotic mixing of the $p-1$ sequences

$$(x_0^1, x_1^1, x_2^1, \ldots, x_n^1, x_{n+1}^1, \ldots),$$
$$(x_0^2, x_1^2, x_2^2, \ldots, x_n^2, x_{n+1}^2, \ldots), \ldots,$$
$$(x_0^{p-1}, x_1^{p-1}, x_2^{p-1}, \ldots, x_n^{p-1}, x_{n+1}^{p-1}, \ldots), \ldots.$$

Using the last one $(x_0^p, x_1^p, x_2^p, \ldots, x_n^p, x_{n+1}^p, \ldots)$ in order to distribute the iterated points with respect to this given partition defining the subsequence $(\overline{x_0}, \overline{x_1}, \overline{x_2}, \ldots, \overline{x_q}, \overline{x_{q+1}}, \ldots)$ by

$$\begin{cases} n_{(-1)} = -1 \\ \overline{x_q} = x_{n_{(q)}}^k, \quad \text{with } n_{(q)} = \min\limits_{1 \leq k \leq p-1} \left\{ s_k(q) = \min\limits_{r \in \mathbb{N}} \{ r_k > n_{(q-1)} \,|\, x_{r_k}^p \in J_k \} \right\}. \end{cases} \tag{9}$$

The pseudo-code, for computing the iterates of (9) corresponding to N iterates of (3) is:

$X_0 = (x_0^1, x_0^2, \ldots, x_0^{p-1}, x_0^p) = \text{seed}$
$n = 0;\ q = 0\ ;$
do { while $n < N$
do {while $(x_n^p \in J_0)$ compute
$(x_n^1, x_n^2, \ldots, x_n^{p-1}, x_n^p); n{+}{+}\}$
compute $(x_n^1, x_n^2, \ldots, x_n^{p-1}, x_n^p)$
let k be such that $x_n^p \in J_k$
then $n(q) = n; \overline{x_q} = x_{n(q)}^k; n{+}{+}; q{+}{+}\}$

Remark 2.2. In this case also, $N\,\mathrm{Sampl}_{\mathrm{iter}}$ is not known *a priori*, however, considering that the selecting process is linked to the uniform distribution of the iterates of the tent map on J, one has

$$N\,\mathrm{Sampl}_{\mathrm{iter}} \approx \frac{2N}{1 - T_1}.$$

Remark 2.3. This second mechanism is more or less linked to the whitening process [Viega, 2003; Viega & Messier, 2003].

Remark 2.4. Actually, one can choose any of the components in order to sample and mix the sequence, not only the last one.

2.3. *Double threshold chaotic sampling*

One can eventually improve the CPRG, previously introduced, with respect to the infinity norm instead of the L_1 or L_2 norms because the L_∞ norm is more sensitive than the others to reveal the concealed f [Lozi, 2009]. For this purpose we introduce a second kind of threshold $T' \in \mathbb{N}$, together with $T_1, \ldots, T_{p-1} \in J$ such that the subsequence $(\overline{x_0}, \overline{x_1}, \overline{x_2}, \ldots, \overline{x_q}, \overline{x_{q+1}}, \ldots)$ is defined by

$$\begin{cases} n_{(-1)} = -1 \\ \overline{x_q} = x_{n_{(q)}}^k, \quad \text{with } n_{(q)} = \min\limits_{1 \leq k \leq p-1} \left\{ s_k(q) = \min\limits_{r_k \in \mathbb{N}} \{ r_k > n_{(q-1)} + T' \,|\, x_{r_k}^p \in J_k \} \right\}. \end{cases} \tag{10}$$

In pseudo-code (10) is then:

$X_0 = (x_0^1, x_0^2, \ldots, x_0^{p-1}, x_0^p) = \text{seed}$
$n = 0;\ q = 0;$
`do {` `while` $n < N$
`do {while` $(n \leq n_{(q-1)} + T'$ `and` $x_n^p \in J_0)$
`compute` $(x_n^1, x_n^2, \ldots, x_n^{p-1}, x_n^p); n$`++``}`
`compute` $(x_n^1, x_n^2, \ldots, x_n^{p-1}, x_n^p)$
`let` k `be such that` $x_n^p \in J_k$
`then` $n(q) = n; \overline{x_q} = x_{n(q)}^k; n$`++`$; q$`++``}`

Remark 2.5. In this case also, $N\,\mathrm{Sampl}_{\mathrm{iter}}$ is not known *a priori*, it is more complicated to give an equivalent to it. However, considering that the selecting process is linked to the uniform distribution of the iterates of the tent map on J, and to the second threshold T', it implies that

$$N\,\mathrm{Sampl}_{\mathrm{iter}} \leq \min\left\{\frac{2N}{1-T_1}, \frac{N}{T'}\right\}.$$

Remark 2.6. The second kind of threshold T' can also be used with only the chaotic sampling, without the chaotic mixing.

3. Emergence of Randomness

Numerical results on chaotic numbers produced by (3)–(9) show that they are equally distributed over the interval J with a very good precision [Lozi, 2008a, 2008b, 2009]. (See also Appendix A.2.)

In this section we emphasize that when the parameters $\epsilon_{i,j}$ belong to a special window (called the window of emergence), the M-p CPRNG defined above behaves well.

3.1. *Approximated invariant measures*

In order to perform numerical computation, we have to define some numerical tools — the approximated invariant measures.

First we define an approximation $P_{M,N}(x)$ of the invariant measure also called the probability distribution function linked to the 1-dimensional map f when computed with floating numbers (or numbers in double precision). In this scope we consider a regular partition of M small intervals (boxes) r_i of J defined by

$$s_i = -1 + \frac{2i}{M}, \quad i = 0, M \tag{11}$$

$$r_i = [s_i, s_{i+1}[, \quad i = 0, M-2 \tag{12}$$

$$r_{M-1} = [s_{M-1}, 1] \tag{13}$$

$$J = \bigcup_0^{M-1} r_i \tag{14}$$

the length of each box is

$$s_{i+1} - s_i = \frac{2}{M} \tag{15}$$

(note that this regular partition of J is different from the previous one linked to the threshold values T_i, according to (8)).

All iterates $f^{(n)}(x)$ belonging to these boxes are collected (after a transient regime of Q iterations decided *a priori*, i.e. the first Q iterates are neglected). Once the computation of $N + Q$ iterates is completed, the relative number of iterates with respect to N/M in each box r_i represents the value $P_N(s_i)$. The approximated $P_N(x)$ defined in this article is then a step function, with M steps. As M may vary, we define

$$P_{M,N}(s_i) = \frac{M}{N}(\#r_i) \tag{16}$$

where $\#r_i$ is the number of iterates belonging to the interval r_i. $P_{M,N}(x)$ is normalized to 2 on the interval J.

$$P_{M,N}(x) = P_{M,N}(s_i) \quad \forall\, x \in r_i. \tag{17}$$

In the case of p-coupled maps, we are more interested by the distribution of each component $(x^1, x^2, x_2^1, \ldots, x^p)$ of X rather than the distribution of the variable X itself in J^p. We then consider the approximated probability distribution function $P_{M,N}(x^j)$ associated to one among several components of $F(X)$ defined by (3) which are one-dimensional maps. In this paper, we use equally N_{disc} for M and N_{iter} for N when they are more explicit.

The discrepancies E_1 (in norm L_1), E_2 (in norm L_2) and E_∞ (in norm L_∞) between $P_{N_{\mathrm{disc}},N_{\mathrm{iter}}}(x)$ and the Lebesgue measure which is the invariant measure associated to the symmetric tent map, are defined by

$$E_{1,N_{\mathrm{disc}},N_{\mathrm{iter}}}(x) = \|P_{N_{\mathrm{disc}},N_{\mathrm{iter}}}(x) - 1\|_{L_1} \tag{18}$$

$$E_{2,N_{\mathrm{disc}},N_{\mathrm{iter}}}(x) = \|P_{N_{\mathrm{disc}},N_{\mathrm{iter}}}(x) - 1\|_{L_2} \tag{19}$$

$$E_{\infty,N_{\mathrm{disc}},N_{\mathrm{iter}}}(x) = \|P_{N_{\mathrm{disc}},N_{\mathrm{iter}}}(x) - 1\|_{L_\infty}. \tag{20}$$

In the same way, an approximation of the correlation distribution function $C_{M,N}(x, y)$ is obtained

numerically building a regular partition of M^2 small squares (boxes) of J^2 imbedded in the phase subspace (x^l, x^m).

$$s_i = -1 + \frac{2i}{M}, \quad t_j = -1 + \frac{2j}{M}, \quad i,j = 0, M \tag{21}$$

$$r_{i,j} = [s_i, s_{i+1}[\times [t_j, t_{j+1}[, \quad i,j = 0, M-2 \tag{22}$$

$$r_{M-1,j} = [s_{M-1}, 1] \times [t_j, t_{j+1}[, \quad j = 0, M-2 \tag{23}$$

$$r_{i,M-1} = [s_i, s_{i+1}[\times [t_{M-1}, 1], \quad i = 0, M-2 \tag{24}$$

$$r_{M-1,M-1} = [s_{M-1}, 1] \times [t_{M-1}, 1] \tag{25}$$

the measure of the area of each box is

$$(s_{i+1} - s_i)(t_{i+1} - t_i) = \left(\frac{2}{M}\right)^2. \tag{26}$$

Once $N+Q$ iterated points (x_n^1, x_n^m) belonging to these boxes are collected, the relative number of iterates with respect to N/M^2 in each box $r_{i,j}$ represents the value $C_N(s_i, t_j)$. The approximated probability distribution function $C_N(x,y)$ defined here is then a 2-dimensional step function, with M^2 steps. As M can take several values in the next sections, we define

$$C_{M,N}(s_i, t_j) = \frac{M^2}{N}(\#r_{i,j}) \tag{27}$$

where $\#r_{i,j}$ is the number of iterates belonging to the square $r_{i,j}$. $C_{M,N}(x,y)$ is normalized to 4 on the square J^2.

$$C_{M,N}(x,y) = C_{M,N}(s_i, t_j) \quad \forall (x,y) \in r_{i,j}. \tag{28}$$

The discrepancies E_{C_1} (in norm L_1), E_{C_2} (in norm L_2) and E_{C_∞} (in norm L_∞) between $C_{N_{\rm disc},N_{\rm iter}}(x,y)$ and the uniform distribution on the square, are defined by

$$E_{C_1,N_{\rm disc},N_{\rm iter}}(x,y) = \|C_{N_{\rm disc},N_{\rm iter}}(x,y) - 1\|_{L_1} \tag{29}$$

$$E_{C_2,N_{\rm disc},N_{\rm iter}}(x,y) = \|C_{N_{\rm disc},N_{\rm iter}}(x,y) - 1\|_{L_2} \tag{30}$$

$$E_{C_\infty,N_{\rm disc},N_{\rm iter}}(x,y) = \|C_{N_{\rm disc},N_{\rm iter}}(x,y) - 1\|_{L_\infty}. \tag{31}$$

Finally let $AC_{N_{\rm disc},N_{\rm iter}}(x,y)$ be the autocorrelation distribution function which is the correlation function $C_{N_{\rm disc},N_{\rm iter}}(x,y)$ of (28) defined in the phase space (x_n^l, x_{n+1}^l) instead of the phase space (x^l, x^m). In order to control that the enhanced chaotic numbers $(\overline{x_0}, \overline{x_1}, \overline{x_2}, \ldots, \overline{x_q}, \overline{x_{q+1}}, \ldots)$ are uncorrelated, we plot them in the phase subspace $(\overline{x_q}, \overline{x_{q+1}})$ and we check if they are uniformly distributed in the square J^2 and if f is concealed (i.e. $E_{AC_1,N_{\rm disc},N_{\rm iter}}(\overline{x_q}, \overline{x_{q+1}})$, $E_{AC_2,N_{\rm disc},N_{\rm iter}}(\overline{x_q}, \overline{x_{q+1}})$, $E_{AC_\infty,N_{\rm disc},N_{\rm iter}}(\overline{x_q}, \overline{x_{q+1}})$ vanish).

3.2. *A window of emergence of randomness*

In order to point out the usefulness of the double threshold chaotic sampling, we simply consider the case of only 4-coupled equation, and such that:

$$\epsilon_{i,j} = \epsilon_i \quad \forall\, i \neq j \quad \text{and} \quad \epsilon_{i,i} = 1 - 3\epsilon_i \tag{32}$$

Eq. (3) becomes (33):

$$\begin{cases} x_{n+1}^1 = (1-3\epsilon_1) f(x_n^1) + \epsilon_1 f(x_n^2) \\ \qquad\quad + \epsilon_1 f(x_n^3) + \epsilon_1 f(x_n^4) \\ x_{n+1}^2 = \epsilon_2 f(x_n^1) + (1-3\epsilon_2) f(x_n^2) \\ \qquad\quad + \epsilon_2 f(x_n^3) + \epsilon_2 f(x_n^4) \\ x_{n+1}^3 = \epsilon_3 f(x_n^1) + \epsilon_3 f(x_n^2) \\ \qquad\quad + (1-3\epsilon_3) f(x_n^3) + \epsilon_3 f(x_n^4) \\ x_{n+1}^4 = \epsilon_4 f(x_n^1) + \epsilon_4 f(x_n^2) \\ \qquad\quad + \epsilon_4 f(x_n^3) + (1-3\epsilon_4) f(x_n^4) \end{cases} \tag{33}$$

Moreover we assume that

$$\epsilon_i = i\epsilon_1. \tag{34}$$

For the sake of simplicity we consider only the chaotic sampling method (i.e. we use only one threshold T), without the chaotic mixing. We then compute $E_{1,N_{\rm disc},N_{\rm iter}}(\overline{x}), E_{2,N_{\rm disc},N_{\rm iter}}(\overline{x})$, $E_{\infty,N_{\rm disc},N_{\rm iter}}(\overline{x})$ and $E_{AC_1,N_{\rm disc},N_{\rm iter}}(\overline{x_q}, \overline{x_{q+1}})$, $E_{AC_2,N_{\rm disc},N_{\rm iter}}(\overline{x_q}, \overline{x_{q+1}})$, $E_{AC_\infty,N_{\rm disc},N_{\rm iter}}(\overline{x_q}, \overline{x_{q+1}})$ for $N_{\rm disc} = 1024$ and $N_{\rm iter} = 10^{11}$. We choose $T = 0.9$ and $T' = 20$. We display in Fig. 1 the values of the six computed errors when $\epsilon_1 \in [10^{-17}, 10^{-1}]$, the seed (initial values) being $x_0^1 = 0.330000$, $x_0^2 = 0.338756$, $x_0^3 = 0.504923$, $x_0^4 = 0.324082$.

A window of emergence comes clearly into sight for the values $\epsilon_1 \in [10^{-15}, 10^{-7}]$ if one considers all together the six errors.

The errors $E_{\infty,N_{\rm disc},N_{\rm iter}}(\overline{x})$ and $E_{AC_\infty,N_{\rm disc},N_{\rm iter}}(\overline{x_q}, \overline{x_{q+1}})$ narrowing this window in which $340\,753\,095 \leq N\,{\rm Sampl}_{\rm iter} \leq 340\,768\,513$ out of $N_{\rm iter} = 10^{11}$.

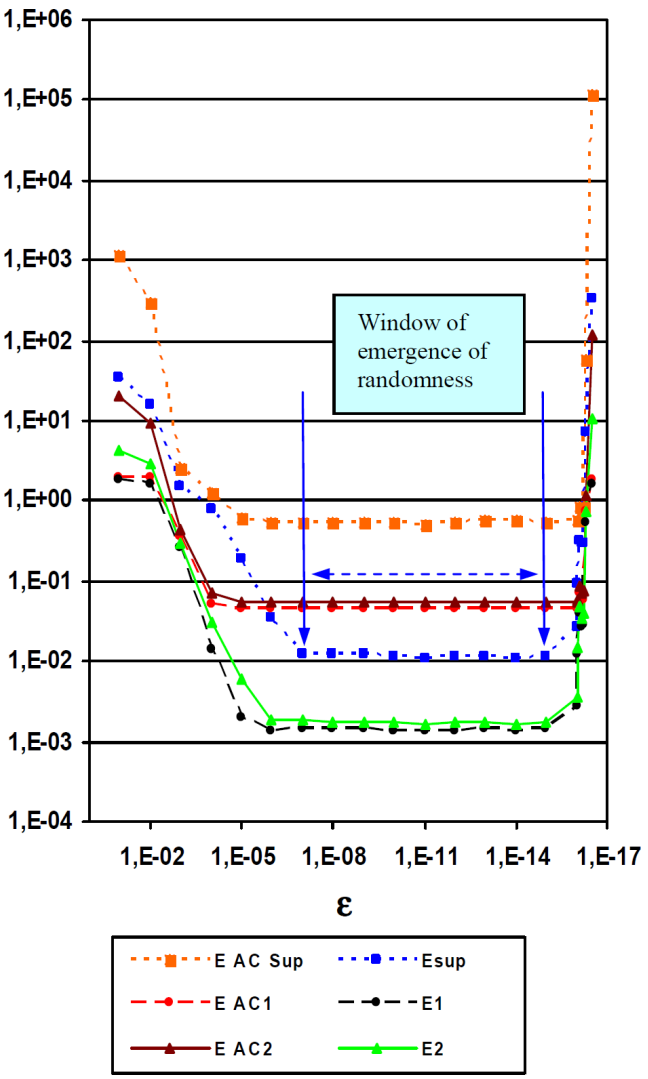

Fig. 1. The window of emergence of randomness.

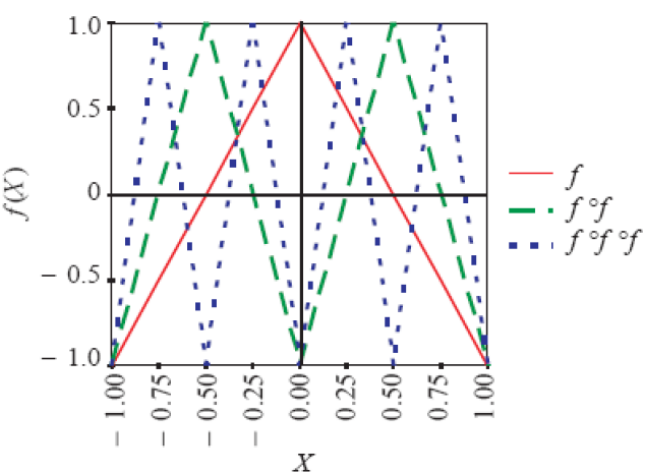

Fig. 2. Graphs of the symmetric tent map $f, f^{(2)}$ and $f^{(3)}$ on the interval $[-1, 1]$.

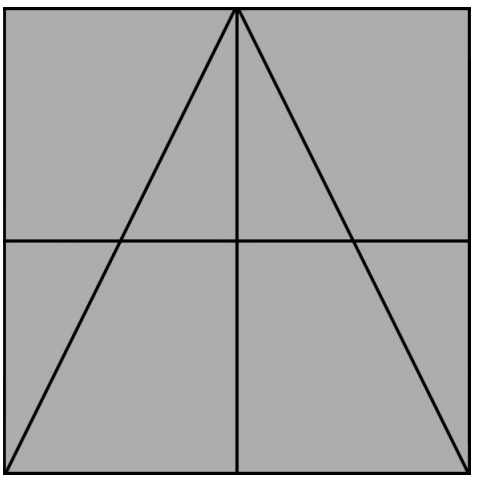

Fig. 3. In shaded regions the autocorrelation distribution $AC_{M,N}(x, y)$ is constant for the symmetric tent map f on the interval $[-1, 1]$ for $M = 1$ or 2.

3.3. *The underneath of randomness*

The double threshold chaotic sampling is very efficient because its aim is mainly to conceal f in the most drastic way. In order to understand the underneath mechanism, consider first that in the phase space (x_n^l, x_{n+1}^l) the graph of the chaotically sampled chaotic numbers is a mix of the graphs of the $f^{(r)}$ for all $r \in \mathbb{N}$ (Fig. 2).

It is obvious as shown in Fig. 3 that for $r = 1$ if $M = 1$ or 2, $AC_{M,N}(x, y)$ is constant and normalized on the square hence $E_{AC_1,N_{\text{disc}},N_{\text{iter}}}(x, y) = E_{AC_2,N_{\text{disc}},N_{\text{iter}}}(x, y) = E_{AC_\infty,N_{\text{disc}},N_{\text{iter}}}(x, y) = 0$.

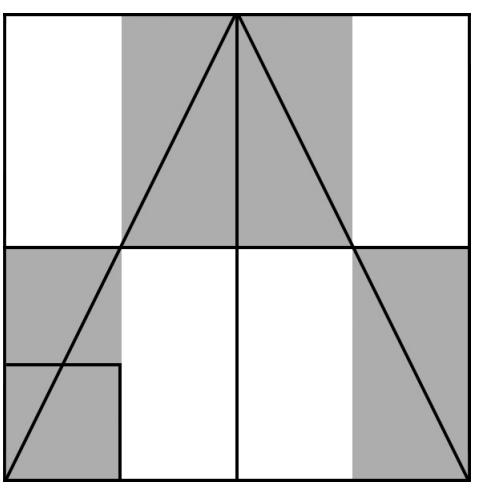

Fig. 4. Regions where the autocorrelation distribution $AC_{M,N}(x, y)$ is constant for the symmetric tent map f are shaded, for $M = 4$. (The square on the bottom left-hand side of the graph shows the size of the $r_{i,j}$ box.) $AC_{M,N}(x, y)$ vanishes on the white regions.

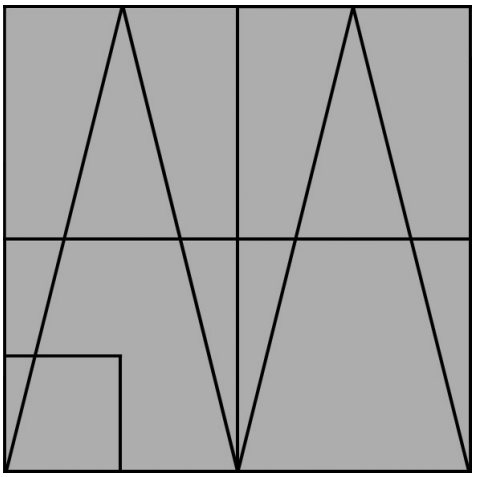

Fig. 5. In shaded regions the autocorrelation distribution $AC_{M,N}(x,y)$ is constant for the symmetric tent map $f^{(2)}$ on the interval $[-1,1]$ for $M = 1, 2$ and 4.

The autocorrelation function is different from zero only if $M > 2$ (Fig. 4).

In the same way as displayed in Figs. 5–7, $E_{AC_1,N_{\rm disc},N_{\rm iter}}(x,y) = E_{AC_2,N_{\rm disc},N_{\rm iter}}(x,y) = E_{AC_\infty,N_{\rm disc},N_{\rm iter}}(x,y) = 0$ for $f^{(i)}$ iff $M < 2^i$. Hence for a given M, if we cancel the contribution of all $f^{(i)}$ for $2^i < M$, it is not possible to identify the genuine function f.

3.4. *Testing the randomness*

As shown previously [Lozi, 2008a] (see also Appendices A.2 and A.3), the errors in L_1 or L_2 norms decrease with the number of chaotic points (as in the law of large numbers) and conversely increase with the number M of boxes used to define

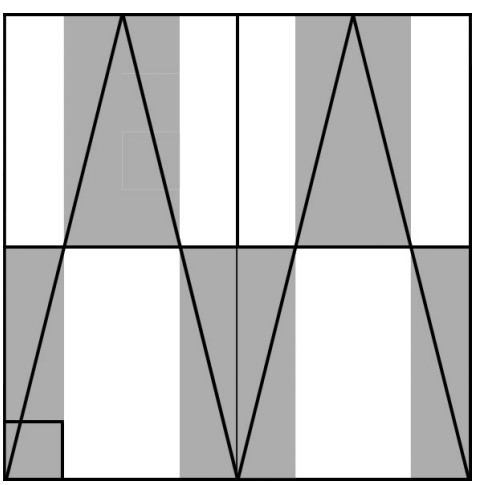

Fig. 6. Regions where the autocorrelation distribution $AC_{M,N}(x,y)$ is constant for the symmetric tent map $f^{(2)}$ are shaded for $M = 8$.

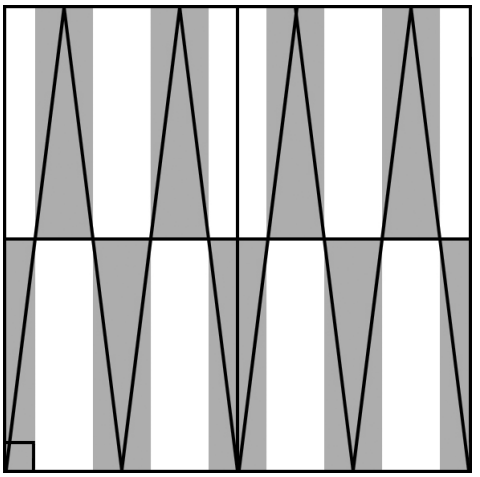

Fig. 7. Regions where the autocorrelation distribution $AC_{M,N}(x,y)$ is constant for the symmetric tent map $f^{(3)}$ are shaded for $M = 16$.

$AC_{M,N}(x,y)$. It is the same for the error in L_∞ norm.

Figure 8 shows that when M is greater than 2^5, the sequence defined by (10) behaves better than the one defined by (6) or (9) when applied to (33).

Figure 9 shows that when the number of chaotic points increases the error $E_{AC_\infty,N_{\rm disc},N_{\rm iter}}(\overline{x}_q, \overline{x}_{q+1})$

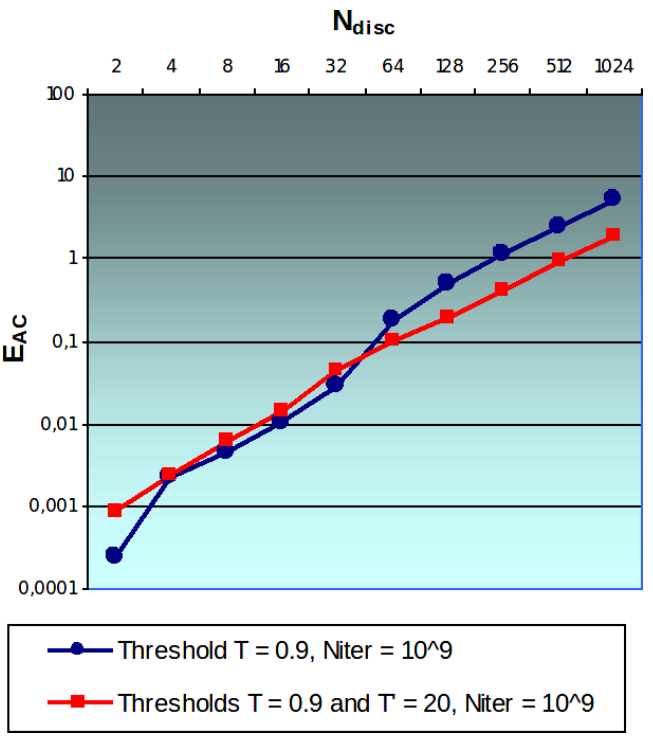

Fig. 8. Error of $E_{AC,N_{\rm disc},N_{\rm iter}}(\overline{x_q}, \overline{x_{q+1}})$, $N_{\rm disc} = 2^1$ to 2^{10}, $N_{\rm iter} = 10^9$, thresholds $T = 0.9$ and $T' = 20$, $\epsilon_i = i\epsilon_1$, $\epsilon_i = 10^{-14}$. Computations are done using double precision numbers (∼14–15 digits).

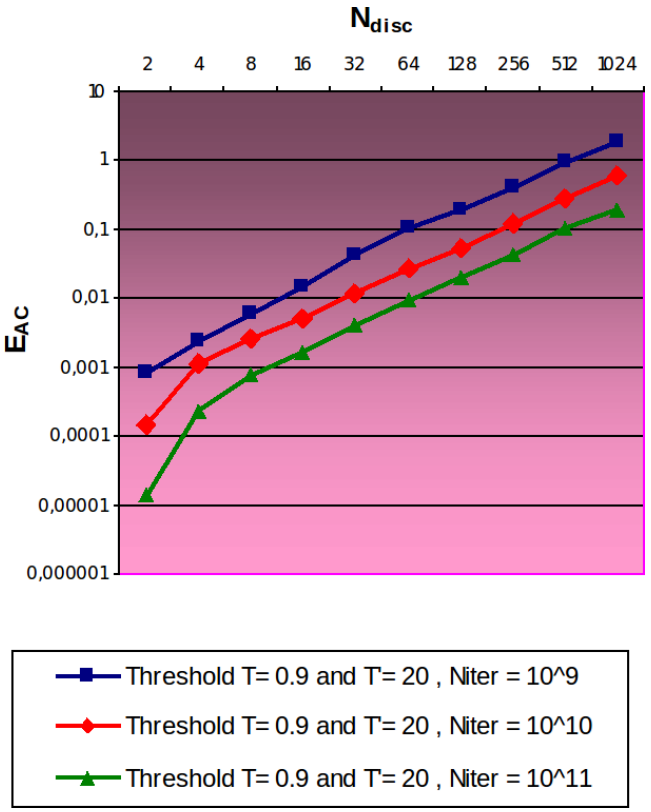

Fig. 9. Error of $E_{AC,N_{\rm disc},N_{\rm iter}}(\overline{x_q},\overline{x_{q+1}})$, $N_{\rm disc}=2^1$ to 2^{10}, $N_{\rm iter}=10^9$ to 10^{11}, thresholds $T=0.9$ and $T'=20$, $\epsilon_i=i\epsilon_1$, $\epsilon_i=10^{-14}$. Computations are done using double precision numbers (∼14–15 digits).

decreases drastically. If for example $T'>100$, it is necessary to use a huge grid of $2^{100}\times 2^{100}$ boxes splitting the square J^2 in order to find a trace of the genuine function f. This is numerically impossible with double precision numbers. Then the chaotic numbers emerge as random numbers.

4. Applications

Generation of random or pseudorandom numbers, nowadays, is a key feature of industrial mathematics. Pseudorandom or chaotic numbers are used in many areas of contemporary technology such as modern communication systems and engineering applications. More and more European or US patents using discrete mappings for this purpose are obtained by researchers of discrete dynamical systems [Petersen & Sorensen, 2007; Ruggiero *et al.*, 2006].

When an efficient M-p CPRNG is defined, there exists a huge number of applications for the pseudorandom numbers it can generate, as for example chaotic masking, chaotic modulation or chaotic shift keying in the fields of secure communications [Hénaff *et al.*, 2009a, 2009b, 2009c, 2010].

4.1. *Parameter sensitivity*

We have improved a determining property of the M-p CPRNG in this paper via Eq. (33) and double threshold chaotic sampling (10) is the high number of parameters used ($p\times(p-1)$ for p coupled equations) which allows to choose it as cipher-keys, however this achievement is possible only if there is a high sensitivity to the parameters values.

In order to point out this sensitivity, it is enough to consider the simplest case of 2-coupled equations with two sets of slightly different parameters (ϵ_1,ϵ_2) and $(\epsilon_1^*,\epsilon_2)$: $\epsilon_1=0.000001$, $\epsilon_1^*=0.0000010000000000003$ and $\epsilon_2=0.000002$.

$$\begin{cases} x_{n+1}^1=(1-\epsilon_1)f(x_n^1)+\epsilon_1 f(x_n^2)\\ x_{n+1}^2=\epsilon_2 f(x_n^1)+(1-\epsilon_2)f(x_n^2)\end{cases}\tag{35}$$

$$\begin{cases} x_{n+1}^{*1}=(1-\epsilon_1)f(x_n^{*1})+\epsilon_1^* f(x_n^{*2})\\ x_{n+1}^{*2}=\epsilon_2 f(x_n^{*1})+(1-\epsilon_2)f(x_n^{*2})\end{cases}\tag{36}$$

The double threshold sampling is done using $T=0.9$ and $T'=20$ and the same seed is taken

$$X_0=(x_0^1,x_0^2)=X_0^*=(x_0^{*1},x_0^{*2}).$$

Despite the fact that the difference between ϵ_1 and ϵ_1^* is tiny: $\frac{|\epsilon_1-\epsilon_1^*|}{\epsilon_1}=3\times10^{-13}$, the sequences $(\overline{x_0},\overline{x_1},\overline{x_2},\ldots,\overline{x_q},\overline{x_{q+1}},\ldots)$ and $(\overline{x_0^*},\overline{x_1^*},\overline{x_2^*},\ldots,\overline{x_q^*},\overline{x_{q+1}^*},\ldots)$ differ completely as displayed in Table 1. (In fact, all the components $(x^1_{n_{(q)}},x^2_{n_{(q)}})$ and $(x^{*1}_{n_{(q)}},x^{*2}_{n_{(q)}})$ are different.)

Then rather than a unique CPRNG which is introduced here, there is a quasi-infinite family of CPRNG that the M-p CPRNG define allowing several possibilities of applications.

4.2. *Gaussian noise*

As an example of such application, the generation of Gaussian noise from the sequences $(\overline{x_0},\overline{x_1},\overline{x_2},\ldots,\overline{x_q},\overline{x_{q+1}},\ldots)$ is very easy when a Box–Muller transform is applied.

A Box–Muller transform [Box & Muller, 1958] is a method of generating pairs of independent standard normally distributed (zero expectation, unit variance) random numbers, given a source of uniformly distributed random numbers. The polar form [Knop, 1969] of such a transform takes two

Table 1. Sequences $(x^1_{n_{(q)}}, x^{*1}_{n_{(q)}})$ and $(x^2_{n_{(q)}}, x^{*2}_{n_{(q)}})$ of Eqs. (35) and (36) with $\epsilon_1 = 0.000001$, $\epsilon^*_1 = 0.0000010000000000003$ and $\epsilon_2 = 0.000002$. $X_0 = (x^1_0, x^2_0) = X^*_0 = (x^{*1}_0, x^{*2}_0)$.

ϵ_1	0.000001	ϵ^*_1	0.0000010000000000003
x^1_0	0.330000013113021851	x^{*1}_0	0.330000013113021851
$x^1_{n_{(0)}}$	−0.959214817207605153	$x^{*1}_{n_{(0)}}$	−0.0585367291739744555
$x^1_{n_{(1)}}$	0.657775688600752417	$x^{*1}_{n_{(1)}}$	0.386129403866398935
$x^1_{n_{(2)}}$	−0.784600935471051031	$x^{*1}_{n_{(2)}}$	0.471824729381262631
ϵ_1	0.000001	ϵ^*_1	0.0000010000000000003
x^2_0	0.338756413113021848	x^{*2}_0	0.338756413113021848
$x^2_{n_{(0)}}$	0.914472270898123885	$x^{*2}_{n_{(0)}}$	−0.646249812458326023
$x^2_{n_{(1)}}$	0.9156844129956766	$x^{*2}_{n_{(1)}}$	0.894262910879751405
$x^2_{n_{(2)}}$	0.910813705361448345	$x^{*2}_{n_{(2)}}$	0.820811987022524114

samples from a different interval $[-1, 1]$ and maps them to two normally distributed samples without the use of sine or cosine functions. This form of the polar transform is widely used, in part due to its inclusion in Numerical Recipes.

As the sequences $(\overline{x_0}, \overline{x_1}, \overline{x_2}, \ldots, \overline{x_q}, \overline{x_{q+1}}, \ldots)$ are uniformly distributed in $J = [-1, 1] \subset \mathbb{R}$, the application is straightforward.

4.3. *Hash function*

Another example of application could be the computation of hash function. A hash function is any well-defined procedure or mathematical function that converts a large, possibly variable-sized amount of data into a small one. The values returned by a hash function are called hash values, hash codes, hash sums, checksums or simply hashes.

Hash functions are mostly used to speed up table lookup or data comparison tasks — such as finding items in a database, detecting duplicated or similar records in a large file, finding similar stretches in DNA sequences, and so on.

A hash function may map two or more keys to the same hash value. In many applications, it is desirable to minimize the occurrence of such collisions, which means that the hash function must map the keys to the hash values as evenly as possible. Depending on the application, other properties may be required as well. Although the idea was conceived in the 1950s, the design of good hash functions is still a topic of active research.

Although hash function generally involves integers, one can consider that the application which maps the initial seed $X_0 = (x^1_0, x^2_0, \ldots, x^{p-1}_0, x^p_0)$ into any predetermined term of the sequence $(\overline{x_0}, \overline{x_1}, \overline{x_2}, \ldots, \overline{x_q}, \overline{x_{q+1}}, \ldots)$ is a hash function working on floating point numbers.

We will explore this application in a forthcoming paper.

Others applications show the high-potency of such M-p CPRNG. Due to limitation of this article, they will be published elsewhere.

5. Conclusion

Using a double threshold in order to sample a chaotic sequence, we have improved with respect to the infinity norm the M-p CPRNG previously introduced. When the value of the second threshold T' is greater than 100, it is impossible to find the genuine function used to generate the chaotic numbers. The new M-p CPRNG family is robust versus the choice of the weak parameter of the system for $10^{-15} < \epsilon < 10^{-7}$, allowing the use of this family in several applications as for example producing Gaussian noise, computing hash function or in chaotic cryptography.

References

Aziz-Alaoui, M. A. & Bertelle, C. [2009] *From System Complexity to Emergent Properties* (*Understanding Complex Systems*) (Springer-Verlag, Berlin).

Box, G. E. P. & Muller, M. E. [1958] "A note on the generation of random normal deviates," *Ann. Math. Statist.* **29**, 610–611.

Gora, P., Boyarsky, A., Islam, Md. S. & Bahsoun, W. [2006] "Absolutely continuous invariant measures that cannot be observed experimentally," *SIAM J. Appl. Dyn. Syst.* **5**, 84–90 (electronic).

Hénaff, S., Taralova, I. & Lozi, R. [2009a] "Observers design for a new weakly coupled map function," *Conf. Proc. ICCSA 2009, 3rd Int. Conf. Complex Systems and Applications*, June 29–July 02, eds. Bertelle, C., Liu, X. & Aziz-Alaoui, M. A., pp. 47–50.

Hénaff, S., Taralova, I. & Lozi, R. [2009b] "Statistical and spectral analysis of a newly weakly coupled maps system," *Indian J. Industr. Appl. Math.* **2**, 1–17.

Hénaff, S., Taralova, I. & Lozi, R. [2009c] "Dynamical analysis of a new statistically highly performant deterministic function for chaotic signals generation," *IPACS Open Access Electronic Library, Physics and Control 2009*.

Hénaff, S., Taralova, I. & Lozi, R. [2010] "Exact and asymptotic synchronization of a new weakly coupled maps system," *J. Nonlin. Syst. Appl.* **1**, 87–95.

Knop, R. [1969] "Remark on algorithm 334[G5]: Normal random deviates," *Commun. ACM* **12**, 28.

Lanford III, O. E. [1998] "Some informal remarks on the orbit structure of discrete approximations to chaotic maps," *Experim. Math.* **7**, 317–324.

Lozi, R. [2006] "Giga-periodic orbits for weakly coupled tent and logistic discretized maps," *Modern Mathematical Models, Methods and Algorithms for Real World Systems*, eds. Siddiqi, A. H., Duff, I. S. & Christensen, O. (Anamaya Publishers, New Delhi, India), pp. 80–124.

Lozi, R. [2008a] "New enhanced chaotic number generators," *Indian J. Industr. Appl. Math.* **1**, 1–23.

Lozi, R. [2008b] "Chaotic sampling, very weakly coupling, and chaotic mixing: Three simple synergistic mechanisms to make new families of chaotic pseudo random number generators," *6th EUROMECH Non Linear Dynamics Conf.*, Saint-Petersburg, ENOC 2008, 30 June–4 July 2008, IPACS open Access Electronic Library, pp. 1715–1724.

Lozi, R. [2009] "Chaotic pseudo random number generators via ultra weak coupling of chaotic maps and double threshold sampling sequences," *Conf. Proc. ICCSA 2009, 3rd Int. Conf. Complex Systems and Applications*, June 29–July 02, eds. Bertelle, C., Liu, X. & Aziz-Alaoui, M. A., pp. 20–24.

Petersen, M. V. & Sorensen, H. M. [2007] "Method of generating pseudo-random numbers in an electronic device, and a method of encrypting and decrypting electronic data," United States Patent 7170997.

Ruggiero, D., Mascolo, D., Pedaci, I. & Amato, P. [2006] "Method of generating successions of pseudo-random bits or numbers," United States Patent Application 20060251250.

Sprott, J. C. [2003] *Chaos and Time-Series Analysis* (Oxford University Press, Oxford, UK).

Viega, J. [2003] "Practical random number generation in software," *Proc. 19th Ann. Computer Security Applications Conf.*, pp. 129–140.

Viega, J. & Messier, M. [2003] *Secure Programming Cook Book for C and C++* (O'Reilly, Sebastopol, CA).

Appendix

A.1. *Independency of the chaotic subsequences generated by each component*

One key feature of CPRNG is the use of chaotic numbers themselves in order to do the sampling process. This is possible as the sequences of chaotic numbers produced by each component are independent of the others. In order to control that they are uncorrelated, we compute $E_{C_1,N_{\rm disc},N_{\rm iter}}(x^k, x^l)$, $E_{C_2,N_{\rm disc},N_{\rm iter}}(x^k, x^l)$, and $E_{C_\infty,N_{\rm disc},N_{\rm iter}}(x^k, x^l)$ for $1 \le k \le l \le 4$.

Figure 10 displays the error $E_{C_1,N_{\rm disc},N_{\rm iter}}(x^1, x^2)$ versus the number of iterated points of the approximated correlation function between the first

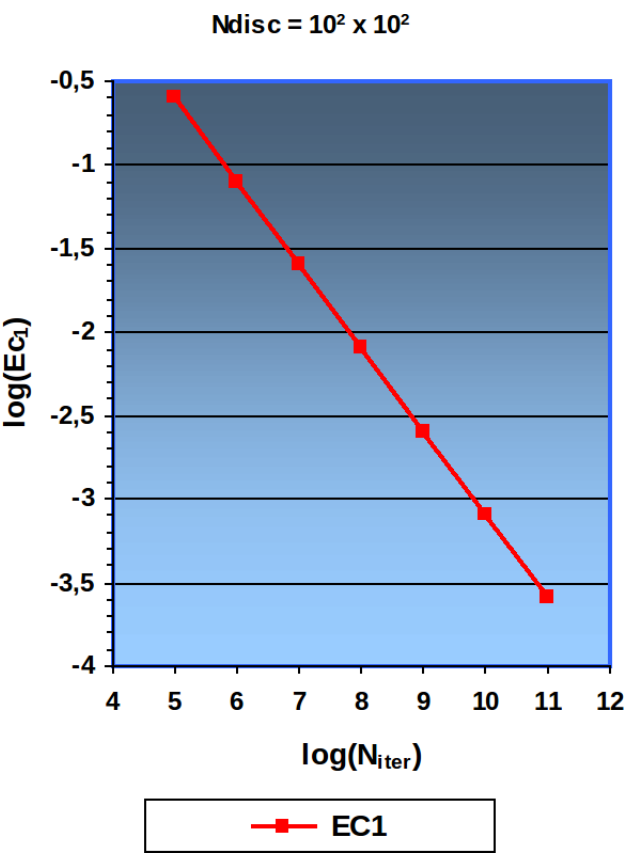

Fig. 10. Error $E_{C_1,N_{\rm disc},N_{\rm iter}}(x^1, x^2)$ for the first and the second components (x^1, x^2) of the 4-coupled symmetric tent map (33). $N_{\rm disc} = 10^2 \times 10^2$, $\epsilon_i = i\epsilon_1$, $\epsilon_i = 10^{-14}$, $N_{\rm iter}$ varies from 10^5 to 10^{11}. Computations are done using double precision numbers (∼14–15 digits). The initial values are $x_0^1 = 0.330$, $x_0^2 = 0.338756$, $x_0^3 = 0.504923$, $x_0^4 = 0.0$.

Table 2. Numerical values corresponding to Fig. 10.

N_{iter}	$E_{C_1,N_{\text{disc}},N_{\text{iter}}}(x^1,x^2)$
10^5	$25\,733\,330 \times 10^{-8}$
10^6	$7\,876\,310 \times 10^{-8}$
10^7	$2\,500\,231 \times 10^{-8}$
10^8	$804\,889 \times 10^{-8}$
10^9	$247\,724 \times 10^{-8}$
10^{10}	$80\,411 \times 10^{-8}$
10^{11}	$26\,640 \times 10^{-8}$

Table 3. Error $E_{C_1,N_{\text{disc}},N_{\text{iter}}}(x^k,x^l)$ for $1 \le k \le l \le 4$ of the 4-coupled symmetric tent map (33). $N_{\text{disc}} = 10^2 \times 10^2$, $N_{\text{iter}} = 10^{11}$, $\epsilon_i = i\epsilon_1$, $\epsilon_i = 10^{-14}$. Computations are done using double precision numbers (∼14–15 digits). The initial values are $x_0^1 = 0.330$, $x_0^2 = 0.338756$, $x_0^3 = 0.504923$, $x_0^4 = 0.0$.

$E_{C_1,N_{\text{disc}},N_{\text{iter}}}$ (x^k,x^l)	$x^l = x^2$	x^3	x^4
$x^k = x^1$	2561×10^{-8}	2551×10^{-8}	2527×10^{-8}
x^2		2522×10^{-8}	2507×10^{-8}
x^2			2486×10^{-8}

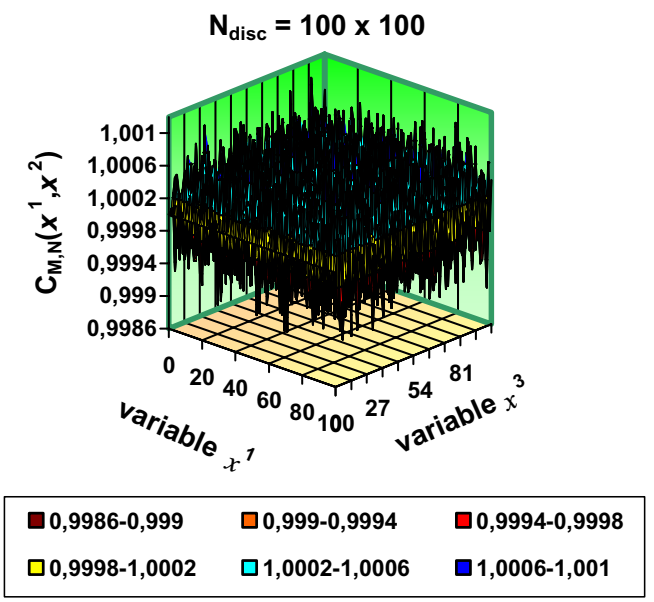

Fig. 11. Difference between the correlation distribution function $C_{N_{\text{disc}},N_{\text{iter}}}(x^1,x^3)$ and the uniform distribution of the 4-coupled symmetric tent map (33). $N_{\text{disc}} = 10^2 \times 10^2$, $N_{\text{iter}} = 10^{11}$, $\epsilon_i = i\epsilon_1$, $\epsilon_i = 10^{-14}$, Computations are done using double precision numbers (∼14–15 digits). The initial values are $x_0^1 = 0.330$, $x_0^2 = 0.338756$, $x_0^3 = 0.504923$, $x_0^4 = 0.0$.

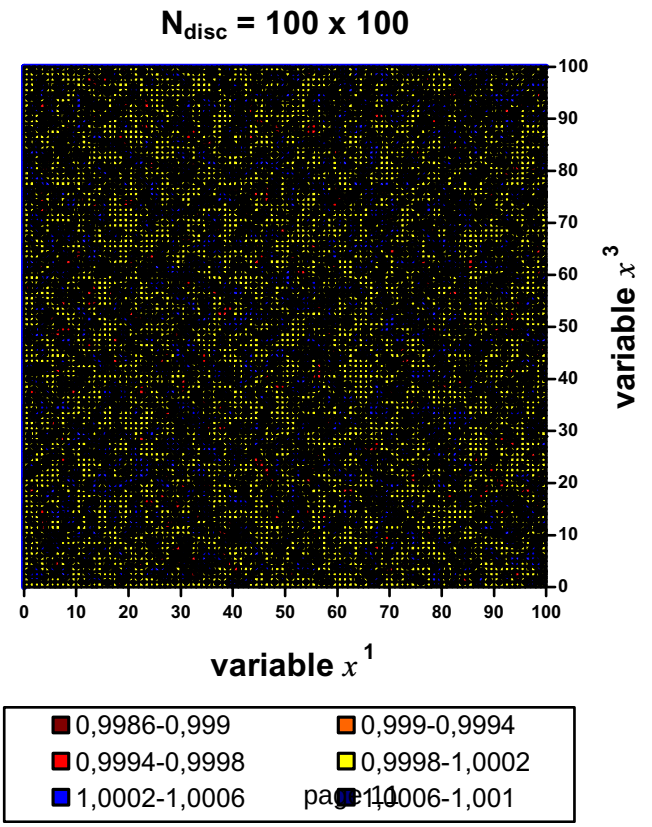

Fig. 12. Projection of Fig. 11 on the phase subspace (x^1,x^3).

and the second components (x^1,x^2) for the 4-coupled symmetric tent map (33). $N_{\text{disc}} = 10^2 \times 10^2$, ϵ_1 is fixed to 10^{-14}, N_{iter} varies from 10^5 to 10^{11}. The corresponding numerical results are displayed in Table 2.

In order to fully verify the uncorrelation, every couple of components must be checked simultaneously. In the considered case $N_{\text{iter}} = 10^{11}$ for the 4-coupled symmetric tent map, the errors $E_{C_1,N_{\text{disc}},N_{\text{iter}}}(x^k,x^l)$ for $1 \le l \le l \le 4$ are displayed in Table 3.

The difference between the correlation distribution function $C_{N_{\text{disc}},N_{\text{iter}}}(x^1,x^3)$ and the uniform distribution of the 4-coupled symmetric tent map is plotted in Fig. 11 and its projection on the phase subspace (x^1,x^3) is displayed in Fig. 12.

A.2. *Distribution of iterates of 4-coupled symmetric tent maps*

We consider the distribution of the iterates of Eq. (33) on the interval $J = [-1,1] \subset \mathbb{R}$. The numerical experiments are performed on several

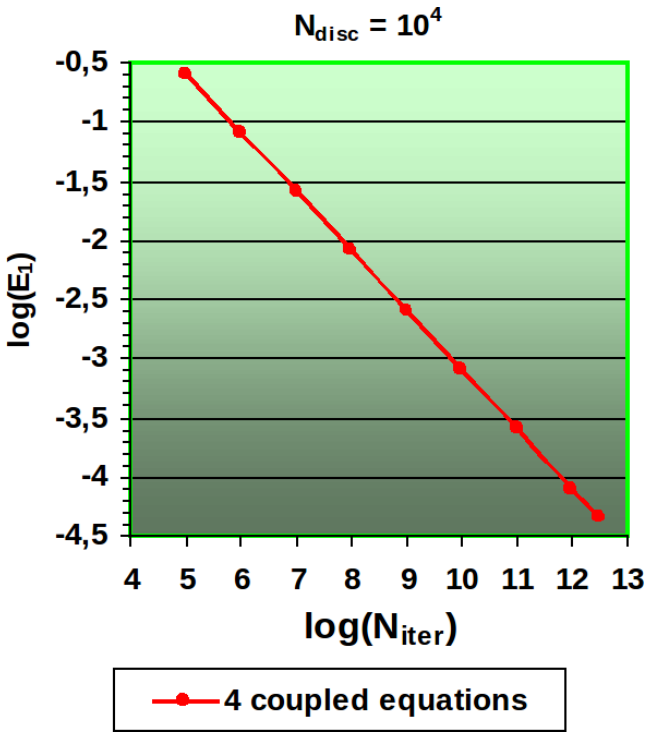

Fig. 13. Error $E_{1,N_{\rm disc},N_{\rm iter}}(x^1)$ of Eq. (33). $N_{\rm disc} = 10^{-4}$, $\epsilon_i = 10^{-14}$, $N_{\rm iter}$ varies from 10^5 to 3×10^{12}. The initial values are $x_0^1 = 0.3300$, $x_0^2 = 0.3387$, $x_0^3 = 0.3313$, $x_0^4 = 0.3332$.

computers involving different microprocessors of Advanced Micro Devices (AMD) and Intel (Centrino and dual core) technologies in order to check the portability of the algorithms we propose. In the same goal the package is written using many versions of Borland C. All the experiments give similar results.

Double precision numbers are used. We fix $\epsilon_1 = 10^{-14}$ in order to belong to the window of emergence (Fig. 1).

Table 4. Numerical values corresponding to Fig. 13.

$N_{\rm iter}$	$E_{1,N_{\rm disc},N_{\rm iter}}(x^3)$
10^5	$24\,991.33 \times 10^{-5}$
10^6	8073.91×10^{-5}
10^7	2526.63×10^{-5}
10^8	807.72×10^{-5}
10^9	256.29×10^{-5}
10^{10}	$79\,701.99 \times 10^{-8}$
10^{11}	$25\,241.40 \times 10^{-8}$
10^{12}	7880.34×10^{-8}
3×10^{12}	4531.71×10^{-8}

As intuitively expected, the density of iterates of each component of (33) converges towards the Lebesgue measure when $\epsilon_1 \to 0$.

The asymptotic properties of dynamical systems intuitively imply that for a fixed value of $N_{\rm disc}$ when the number $N_{\rm iter}$ increases, $E_{1,N_{\rm disc},N_{\rm iter}}(x)$ which measures the discrepancy between $P_{N_{\rm disc},N_{\rm iter}}(x)$ and the Lebesgue measure converges towards 0, except if there exist one or many periodic orbits of finite length lower than $N_{\rm iter}$ which capture the iterates. In this case whatsoever the value of $N_{\rm iter}$ is, the approximated distribution function converges to the distribution function of the periodic orbit if it is unique or to some average

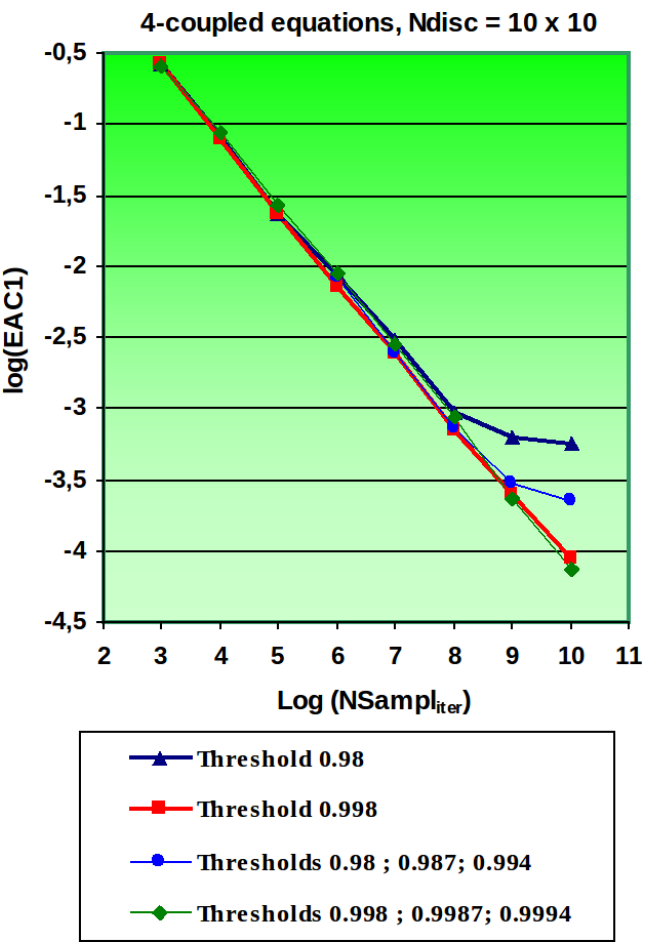

Fig. 14. Error of $E_{AC_1,N_{\rm disc},N_{\rm iter}}(\overline{x_q}, \overline{x_{q+1}})$ for a system of 4-coupled equations when the first component x^1 is sampled by x^4 for both the threshold values 0.98 and 0.998 and when the three components x^1, x^2, x^3 are mixed and sampled by x^4 for the threshold values $T_1 = 0.98$, $T_2 = 0.987$, $T_3 = 0.994$ and $T_1 = 0.998$, $T_2 = 0.9987$, $T_3 = 0.9994$, $N_{\rm disc} = 10 \times 10$, $\epsilon_i = 10^{-14}$, $\epsilon_i = i\epsilon_1$, $N\,{\rm Sampl}_{\rm iter}$ varies from 10^3 to 10^{10}. Initial values: $x_0^1 = 0.3300$, $x_0^2 = 0.3387$, $x_0^3 = 0.3313$, $x_0^4 = 0.3332$.

Table 5. Numerical values corresponding to Fig. 14.

N_{iter}	$N\,\text{Sampl}_{\text{iter}}$	$E_{AC_1,N_{\text{disc}},N_{\text{iter}}}(\overline{x_q},\overline{x_{q+1}})$ 4-Coupled Equations $T=0.998$	$N\,\text{Sampl}_{\text{iter}}$	$E_{AC_1,N_{\text{disc}},N_{\text{iter}}}(\overline{x_q},\overline{x_{q+1}})$ 4-Coupled Equations $T_1=0.998$, $T_2=0.9987$, $T_3=0.9994$
10^5	95	0.70947368	93	0.68924731
10^6	971	0.26570546	1015	0.25881773
10^7	10 095	0.079871223	10 139	0.086706776
10^8	100 622	0.023190157	100 465	0.026815309
10^9	1 001 408	0.0071386288	1 000 549	0.0089111078
10^{10}	9 998 496	0.002493667	9 998 814	0.0027932033
10^{11}	100 013 867	0.00071561417	100 001 892	0.00085967214
10^{12}	999 994 003	0.00025442753	999 945 728	0.000234685100
10^{13}	10 000 042 552	0.000088445108	10 000 046 137	0.000073234736

of the distribution functions of the periodic orbits observed if there are several ones.

Figure 13 shows the errors $E_{1,N_{\text{disc}},N_{\text{iter}}}(x^1)$ versus the number of iterates of the approximated distribution functions with respect to the first variable x^1 for Eq. (33). N_{disc} is fixed to 10^{-4}, $\epsilon_1 = 10^{-14}$, N_{iter} varies from 10^5 to 3×10^{12}. The corresponding numerical results are displayed in Table 4.

A.3. *Comparisons between different sets of parameter values*

In this subsection, we compare the numerical results of method (6) (chaotic sampling) when the threshold values are 0.98 and 0.998 with respect to the auto correlation function $E_{AC_1,N_{\text{disc}},N_{\text{iter}}}(\overline{x_q},\overline{x_{q+1}})$ applied to Eq. (33). In the same figure (Fig. 14) we display the results for both methods (6) and (9) (chaotic sampling and mixing) for the threshold values $T_1 = 0.98$, $T_2 = 0.987$, $T_3 = 0.994$ and $T_1 = 0.998$, $T_2 = 0.9987$, $T_3 = 0.9994$.

In order not to be influenced by the number of iterates which are computed, we compare these results versus the number $N\,\text{Sampl}_{\text{iter}}$ of pseudo random numbers computed which varies upon the values of the thresholds.

A.4. *Impact of the initial values on the results*

It is well known that the choice of the seed of a PRNG is very important. Some seed can lead to the collapse of the period of the computed random numbers. In order to check if the choice of the initial condition of a CPRNG (equivalent to the choice of the seed of a PRNG) changes dramatically the results, we have tested a sequence of different initial values.

Figure 15 shows the distribution of the error $E_{1,N_{\text{disc}},N_{\text{iter}}}(x^1)$ for 500 000 initial values for 4-coupled symmetric tent maps. The computations

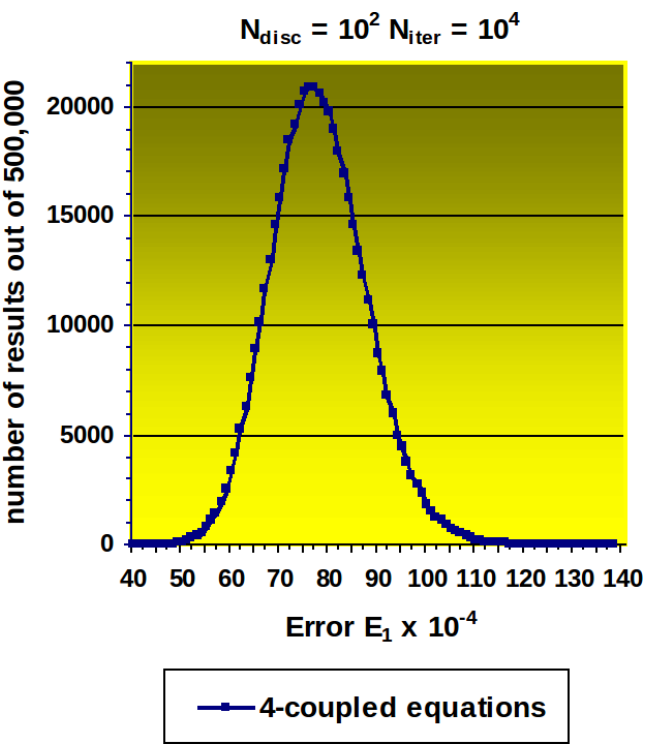

Fig. 15. Distribution of the error $E_{1,N_{\text{disc}},N_{\text{iter}}}(x^1)$ for 500 000 initial values for 4-coupled symmetric tent maps (33). Computations done using double precision numbers (∼14–15 digits), $\epsilon_i = i\epsilon_1$, $\epsilon_i = 10^{-14}$, $N_{\text{iter}} = 10^6$, $N_{\text{disc}} = 10^2$. The initial values are selected following: $x^1_{0,k} = -0.92712 + 10^{-7}\times k$, $x^2_{0,k} = -0.9183636 + 10^{-7}\times 7k$, $x^3_{0,k} = -0.92576657 + 10^{-7}\times 13k$, $x^4_{0,k} = -0.92390643 + 10^{-7}\times 17k$, for $k = 1$ to 500 000.

Table 6. Minimal and maximal values of the $E_{1,N_{\rm disc},N_{\rm iter}}(x^1)$ errors for 500 000 initial values for 4-coupled symmetric tent maps. Computations done using double precision numbers (∼14–15 digits), $\epsilon_i = i\epsilon_1$, $\epsilon_i = 10^{-14}$, $N_{\rm iter} = 10^6$, $N_{\rm disc} = 10^2$.The initial values are selected following: $x^1_{0,k} = -0.92712+10^{-7}\times k$, $x^2_{0,k} = -0.9183636+10^{-7}\times 7k$, $x^3_{0,k} = -0.92576657+10^{-7}\times 13k$, $x^4_{0,k} = -0.92390643+10^{-7}\times 17k$, for $k = 1$ to 500 000.

$N_{\rm disc}$	10^2	10^3	10^4
$\min E_{1,N_{\rm disc},N_{\rm iter}}(x^1)$	4002×10^{-6}	$20\,740 \times 10^{-6}$	$75\,152 \times 10^{-6}$
$\max E_{1,N_{\rm disc},N_{\rm iter}}(x^1)$	$13\,872 \times 10^{-6}$	$30\,116 \times 10^{-6}$	$784\,384 \times 10^{-6}$

are done using double precision numbers (∼14–15 digits), $\epsilon_i = i\epsilon_1$, $\epsilon_i = 10^{-14}$, $N_{\rm iter} = 10^6$, $N_{\rm disc} = 10^2$. The initial values are selected following: $x^1_{0,k} = -0.92712 + 10^{-7} \times k$, $x^2_{0,k} = -0.9183636 + 10^{-7} \times 7k$, $x^3_{0,k} = -0.92576657 + 10^{-7} \times 13k$, $x^4_{0,k} = -0.92390643 + 10^{-7} \times 17k$, for $k = 1$ to 500 000.

The distribution follows more or less a Gaussian distribution, maximal and minimal results are displayed in Table 6.

All these results confirm that the families of chaotic attractor we have introduced are robust versus the choice of the initial seed.